

UNDER DEVELOPMENT

Child's play

He may be a bit old for train sets, but David Robinson thinks virtual model railways can teach kids about PCs. And it's just what the local teens need for their ICT course

One good thing came out of last month's trip to Belfast (*Under Development*, *Shopper* 241). While mooching aimlessly round the airport shops I discovered a game – well, a simulation – that's ideal for getting young users into computers. And, wonderful to report, it cost less than a tenner.

The program in question is Create Your Own Model Railway Deluxe from Focus Multimedia. I'm not sure why it's called Deluxe, as there doesn't seem to be a standard edition for the Deluxe to be better than, but what the hell. The Focus website (www.focusmm.co.uk) lists the price as £9.99 but the airport newsagent was selling it for £8.95, which is good news to any Yorkshireman.

TOYS FOR THE BOYS

My previous experiences with cheap games have been rather chequered, but this one installed in a trice and ran perfectly. And it's brilliant. It comes with two simulators: a 'normal' railway one (albeit with a US flavour) and a Wild West edition complete with the kind of fat-funnelled trains that Butch Cassidy and the Sundance Kid used to rob. Both work in the same way but are equipped with different scenery and rolling stock. The Wild West version also has some annoying 'yee-ha!'-type music, although fortunately you can turn this off.

Now you might think that I'm getting a bit old for messing about with toy trains, but it's not for me, you see. Really. Grandson Aidan, who's four-and-a-half (you can't miss off the half – that's very important for playground cred), has been besotted with trains since before he was two. I've built him a Hornby layout with the leftovers from my son Gary's set, which was put in the loft many years ago, plus lots of lovely bargains from eBay, including a Thomas the Tank engine. Some advice: used track bought from eBay is normally OK, but used points are often cream-crackered and have such a nuisance value that you're better off buying new ones. Anyway, the layout is great, but sometimes it has to be put away



and he can't play with it. The virtual train set on my laptop seemed a great substitute. And so it proved.

One damp, dark, dismal winter's day when it was too awful to play in the garden, we started exploring the dozen or so demo layouts. Some run completely automatically, rather like the show layouts in the National Railway Museum at York, but these quickly become boring. Much better are those complete with remote points controls and forward and reverse speeds that you control yourself. If you get the points the wrong way round, the engine gets derailed and damaged in a realistic fashion, and steam pours out of cracks in the boiler, which is a big incentive to get it right.

NOTWORK RAIL

Shortly after our third crash we're joined by Erin, Aidan's sister, who's two (and-a-half). She's at the stage where she copies everything her brother does, so she wants a mouse, too. I give her a spare one that's not attached to anything and she's really happy waving it about, clicking buttons and giving us a running commentary on the points she says she's changing – even though there's no connection between her mouse and what's happening on the screen.

One of the best features is the ability to place a virtual camera on or above the engine. You can change the viewpoint by moving

the mouse to the edge of the screen and zoom in and out using the scroll wheel. If you leave the mouse pointer at the edge of the screen, the view changes, rather like flying a helicopter round and round the train. It's so realistic that it makes Erin sick. This amuses Aidan no end, and he keeps on doing it while she wails. With a sprog balanced precariously on each knee I've not got a free hand to stop him, so the mayhem continues until Mrs R comes in to see what's going on.

Apart from the fun of it, Aidan has improved his use of the mouse and menus and seems to have no problem recognising which controls need a simple click and which need a click, hold and drag. When we advance to building our own layout, he starts learning about planning and that there are rules about object placement that affect how the design will work or, in some cases, not work. If, for instance, you get a power supply point in the wrong place (say on the wrong side of a points loop) then the engine may not respond to the controls – just like a real set. Solving problems such as this will help develop programming, planning and problem-solving skills, which seem strangely lacking in the current crop of teenagers.

NO KIDDING

For years we have taken kids in year 11 (15- and 16-year-olds) on a week's work experience. In the early

years we had some really good ones who looked like going on to glittering careers. More recently, I've been left wondering why we do it. I start each visit with a short interview so we can work out what each pupil will benefit from and pitch the technical content at the right level. However, compared with the 1990s, none of the kids has ever written a program or even a script in their ICT lessons. Nor, they say, have they built a PC or set up a network. When I ask them what they *do* do, the answer is: "We write essays."

In my view you learn how to program partly by studying but mostly by doing – and learning from your mistakes. And, like cooking, painting, gardening or playing a musical instrument, you'll never do it properly just by reading or talking about it. So, at the start of the week, we give them all a pile of bits, a Windows XP CD and a beginner's book on programming. Half-way through the week I ask them the same simple question, which goes like this: There are basically three kinds of loop: a FOR loop, a WHILE loop and a REPEAT loop. Why do we need three? What's the fundamental difference between them?

To help we give them a sheet with pseudo-code examples of each laid out. All the kids are of a (supposedly) decent academic standard but, out of 12 kids in two years, only one was able to answer the question by the end of the week. When shown the answer, all of them have said, "How could I not work that out for myself?"

I'd turn that question into a competition, but regular readers of *Programming Expert* would have a distinct advantage. So, for the best suggestion as to how to stop existing programmers walking off with the prize, there's a copy of Scott Mueller's excellent *Upgrading and Repairing PCs* (ISBN 0789736977) worth £43. **CS**

David Robinson

Software and systems developer

davidr@computershopper.co.uk

